Docket No.: POU920030010US1

Inventor: Patrick J. Meaney et al.

For: **METHOD FOR PROVIDING AN AREA OPTIMIZED BINARY ORTHOGONALITY CHECKER**

APPLICATION FOR UNITED STATES

LETTERS PATENT

"Express Mail" Mailing Label No. EL965409955USUS
Date of Deposit: April 2, 2004

I hereby certify that this paper is being deposited with the United States Postal Service as "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to: Box Patent Application, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Name: Susan L. Nelson

Signature: Susan L. Nelson

INTERNATIONAL BUSINESS MACHINES CORPORATION

Method for providing an Area Optimized Binary Orthogonality Checker

CROSS-REFERENCE TO RELATED APPLICATIONS

TRADEMARKS

[0001] IBM ® is a registered trademark of International Business Machines Corporation, Armonk, New York, U.S.A. S/390, Z900 and z990 and other names used herein may be registered trademarks, trademarks or product names of International Business Machines Corporation or other companies.

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

[0002] This invention relates to minimizing the area of a binary orthogonality checker implemented in static CMOS circuits.

DESCRIPTION OF BACKGROUND

[0003] This invention relates to minimizing the area of a binary orthogonality checker implemented in static CMOS circuits. Binary orthogonality checkers are used in static CMOS logic to check the control inputs to an orthogonal selector, ensuring there is no data collision between inputs. The checker guarantees that no more than one input to the checker has a true logical value during any comparison cycle. Since the primary purpose of the circuit is a checker, the primary concern for the circuit is to reduce the area the checker uses to complete the function. Within IBM, orthogonality checkers compared each combination of inputs with an AND gate, and then ORed all the outputs (see Fig 1.) Thus as described in IBM Patent US5996040 and US6038626 an

orthogonality checker was employed in a scalable selector system for controlling data transfers and routing in a data processing system, comprising a plurality of input data buses coupled to a multiple-bit, multiple bus selector having data, data valid, and an **orthogonality** check outputs and having multiple data input bus ports coupled for receipt of signal from said plurality of input data buses. The system of US5996040 used a multiplicity of said input data buses provide input controls to said multiple-bit, a multiple bus selector to produce a single output data bus signal on a data output bus, and a data valid signal as a control output. The orthogonality check signal controlled data routing logic to allow dataflow to be connected and reconnected without change to control logic. As will be described, for a generalized N input checker completed with a library of gates the solution grows by the squared number for N gates, and thus takes N squared area to implement. As the size of a prior art orthogonality checker grows with the square of the number of inputs, orthogonality checkers with a small number of inputs have a small size. It would be desirable to have a hierarchical checking structure that could be implemented with a smaller area, as well as one which had a smaller logic growth rate as the number of gates increased.

SUMMARY OF THE INVENTION

[0004] This invention provides a way for minimizing the area of a binary orthogonality checker implemented in static CMOS circuits. The area and logic growth rate of the prior art can be reduced by creating an internal hierarchical checking structure. In accordance with our preferred embodiment we have provided a method for minimizing the gate count and area needed to implement an orthogonality checker given a library of logical gates to implement the circuit and the area for each gate in the library. In accordance with our improvement we have provided for determining the optimal mix of

hierarchical levels and inputs to implement a given orthogonality checker to achieve the minimized circuit. An orthogonality checker was employed in a scalable selector system for controlling data transfer and routing in a data processing system, comprising a plurality of input data buses coupled to a multiple-bit, multiple bus selector having data, data valid, and orthogonality check outputs and having multiple data input bus ports coupled for receipt of signal from said plurality of input data buses. for providing binary orthogonality checking by hierarchically combining the checks with smaller numbers of inputs, so that the total check of a large number of inputs is performed with less gates and in a smaller area. In accordance with our preferred embodiment we combine multiple checks into one larger check, providing that the check is performed on each input set, as well as combining an OR of all the inputs to the check. The resulting OR values are then checked for orthogonality, and the results of all the checks are ORed together (see Fig 3.) This structure is extended to multiple hierarchical levels and works with orthogonality checks of any size or implementation. The invention also determines the optimal hierarchical structure for a given technology library and a given number of inputs to check.

[0005] Additional features and advantages are realized through the techniques of the present invention. Other embodiments and aspects of the invention are described in detail herein and are considered a part of the claimed invention. For a better understanding of the invention with advantages and features, refer to the description and to the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] The subject matter which is regarded as the invention is particularly pointed out and distinctly claimed in the claims at the conclusion of the specification. The foregoing and other objects, features, and advantages of the invention are apparent from the following detailed description taken in conjunction with the accompanying drawings in which:

[0007] FIG. 1 illustrates the prior art for an orthogonality checker. The logic ANDs each combination of inputs, then ORs each of the resulting outputs together to produce the final error signal.

[0008] FIG. 2 illustrates a table of the number of two input gates necessary to implement an N input orthogonality checker.

[0009] FIG. 3A illustrates one N=6 example of a hierarchical orthogonality checker, while FIG. 3B shows a NAND implementation.

[0010] FIG. 4 illustrates one example of Boolean equivalence of flat and hierarchical orthogonality checking.

[0011] FIG. 5 illustrates one example of a library table with area, taken from TSMC .25 MOSIS Library.

[0012] FIG. 6 illustrates the results of computations with a library of 2 input gates.

[0013] FIG. 7 illustrates the results of synthesis runs with access to a full standard cell library.

[0014] The detailed description explains the preferred embodiments of the invention, together with advantages and features, by way of example with reference to the drawings.

DETAILED DESCRIPTION OF THE INVENTION

[0015] To minimize the area of the orthogonality checker, this method minimizes the total gate count necessary to implement the orthogonality logic. This minimization is achieved through implementing the orthogonality logic in a hierarchical structure. This method shows the area benefit of the hierarchical structure and identifies the optimal structure for any static CMOS library.

[0016] The basic orthogonality function identifies if two or more binary signals in a group have a logical true value. The IBM system described in the background compared every combination of two signals in the group and performed a logical OR on each of the outputs. While correct, this produces a structure of

$\frac{N(N-1)}{2}+\left(\frac{N(N-1)}{2}\right)-1=N(N-1)-1$ two input gates. As the number of inputs, N, gets larger, this number grows with N squared. By dividing a set of signals into two, or more groups, the orthogonality of the total set can be determined by: determining the orthogonality of each group individually, forming a new group of the OR value of each group, determining the orthogonality of the new group, and producing the final output by ORing each of the resulting orthogonality signals. For example, let N=6, calling the inputs a, b, c, d, e, and f (see Fig 3.) First, the signals are divided into two groups each of size N=3, call them group 1, containing a, b, and c, and group 2, containing d, e, and f. For group 1 two values are found. First G_1, equal to the orthogonality or the three signals in group 1, and O_1, equal to the OR of the three signals in group 2. Then the same signals are computed for group 2, resulting in G_2 and O_2. A new group, group 3, is formed from all the O_n, signals available, and the orthogonality is formed into G_3. The orthogonality of the entire group, a, b, c, d, e, and f is equal to the OR of G_1, G_2, and G_3 (see Fig. 4). Using a library of 2 input AND and OR gates, the prior implementation contains 29 gates, while the new structure contains 17 gates.

[0017] The prior implementation generalized to an N input checker, it will have one AND gate for each combination of two inputs, or $\binom{N}{2}=\frac{N*(N-1)}{2}$ AND gates. This will be followed by an OR gate with the same number of inputs. Assuming the implementation can be completed with a library of gates consisting of only two input AND and two input OR gates, the size of the final solution takes

$\frac{N(N-1)}{2}+\left(\frac{N(N-1)}{2}\right)-1=N(N-1)-1$ gates. This solution grows with N squared

gates, and thus N squared area (see Fig 2.) The binary orthogonal checker by breaking the total circuit into groups allows the total number of gates needed to be reduced and is quite an improvement.

[0018] To determine the ideal hierarchical structure, the following analysis is necessary. First determine the point when a hierarchical structure is smaller than the prior art structure. For example, with a library of 2 input AND and OR gates, with 4 inputs, the prior art checking structure uses 11 gates, while the hierarchical structure uses only 8 gates. The next step in the method is to determine the optimal mix of ORTH2 and ORTH3 checks to combine to make the hierarchical checker. To determine this, examine the trade off of replacing two ORTH3 circuits with three ORTH2 circuits. The cost of the trade equals 2*(ORTH(3)+OR(3)) – 3*(ORTH(2)+OR(2)). The savings you gain by doing this equals (ORTH(J)-ORTH(J-1))+(OR(J+1)-OR(J)), where J is the number of output signals at the next level of hierarchy, and ORTH(N) is the cost of the orthogonality circuit with N inputs, OR(N) is the cost of the or circuit with N inputs. For this library the cost is equal to the number of 2 input gates necessary to implement the circuit. To substitute multiple groups of circuits, we find the total cost equals the original cost function multiplied by the number of groups substituted. The total savings from multiple substitutions equals (ORTH(J)-ORTH(J-M))+(OR(J+1)-OR(J+1-M)), where M is the number of substitutions. The optimal implementation for a hierarchical orthogonality checker is one that maximizes the value of savings minus cost. This can be determined for any orthogonality checker and library using a spreadsheet of values and building valid values.

[0019] For a CMOS library consisting of NAND, NOR, and INV gates, the basic theory from the previous library holds, however, the sizes of each function must be computed based on the area of the standard cells necessary to complete the function. Also in this library the functions can be optimized to reduce the number of inverters necessary.

[0020] From the Table of Figure 2, we can determine the number of two input gates necessary to implement an N input orthogonality checker.

[0021] FIG. 3 illustrates one example of a hierarchical orthogonality checker. In this example of our preferred embodiment we have determined that the number of gates needed for N=6. By breaking the total circuit into groups the total number of gates deeded can be reduced. In the Example of Fig. 3, N=6, and we form two groups of three gates with ORTH(A,B,C) and ORTH(D,E,F). Each groups check is smaller. With OR(A,B,C) and OR(D,E,F) to produce the final result, the orthogonality of the Ors for each group is checked. The final orthogonality is the OR (ORTH(G1,G2) of the output of each orthogonality check which is the input produced for ORTH(A,B,C,D,E,F).

[0022] From this generalization, in the case of N=6, we note we need seven 2-1 AND gates, and five 3-1 OR gates of a total of 7+5*(3-1) area units, which is equal to 17 area units, a number much smaller than the 29 needed without employing the method of grouping.

[0023] FIG. 4 illustrates one example of Boolean equivalence of flat and hierarchical orthogonality checking of the groups ORTH(A,B,C,D) groups, showing ORTH(A+B,C+D).

[0024] FIG. 5 illustrates an example of a library table with area, taken from TSMC .25 MOSIS Library.

[0025] FIG. 6 illustrates the results of computations with a library of 2 input gates.

[0026] FIG. 7 illustrates the results of synthesis runs with access to a full standard cell library.

[0027] We can generalize by first determined that ORTH# is the size of an orthogonality check of #inputs, and we provide OR# as the size of an OR gate with # inputs. Then

ORTH2+OR2=1+1=2. ORTH3+OR3=3+3+(3-1))+3-1)=7. ORTH4+OR4=(6+(6-1))+(4-1)=14. This illustrates a succession of determinations, and the process can continue for the determined number of gates. We note, interesting that since the hierarchical ORTH4+OR4 contains 8 gates, ORTH4 should never be used.

[0028] Here we should note that the illustrations relating to AND gates can be implemented for NAND gates, equivalent to the circuit of FIGURE 3, where a NAND would be implemented for the illustrated AND gate, a NOR for the illustrated OR gate, with the output of the NOR inverted to a final NAND gate. This is equivalent to the circuit of FIG3, and shown as Figure 3B.

[0029] While the preferred embodiment to the invention has been described, it will be understood that those skilled in the art, both now and in the future, may make various improvements and enhancements which fall within the scope of the claims which follow. These claims should be construed to maintain the proper protection for the invention first described.